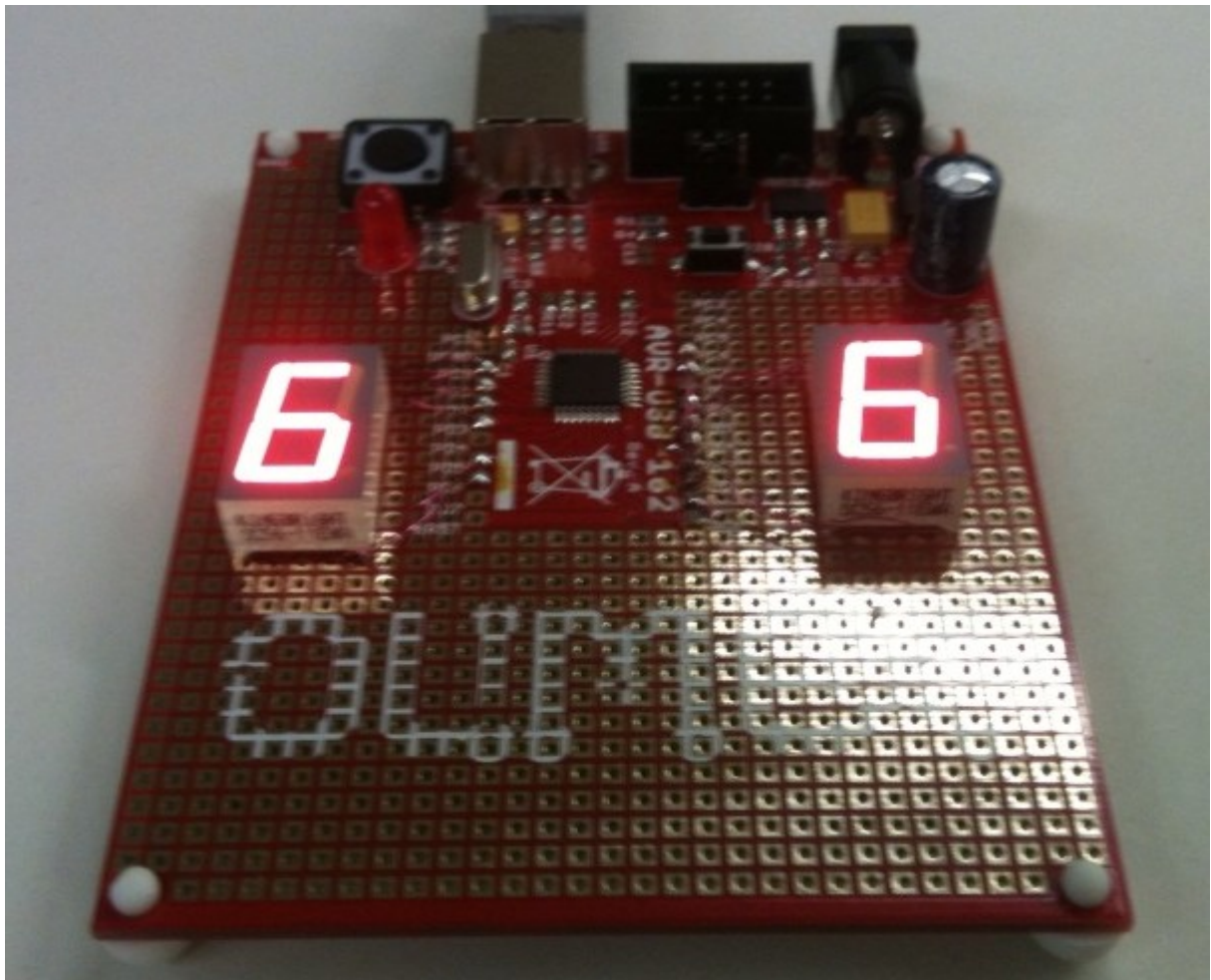


# Projektarbeit „Reaktionsspiel“

aus der  
Datenverarbeitungstechnik



Realisiert auf dem Olimex AVR USB 162 Entwicklerboard

Andreas Loy – Juli 2010

## 1. Aufgabestellung

Die Aufgabe verlangte es ein Reaktionsspiel auf einem Olimex AVR USB 162 zu entwerfen. Auf dem Entwicklerboard Olimex AVR USB 162 stehen zwei 7-Segment-Anzeigen, eine LED und ein Taster zur Verfügung.

## 2. Entwicklungsumgebung

### Hardware:

Olimex AVR USB 162  
2 x 7 Segment Anzeige  
1 x rote LED  
1 x Taster

Der Mikrocontroller AT90USB162 von Atmel bietet 16kB Flash, 512B RAM und 512B EEPROM. Der Chip hat zusätzlich ein USB-Interface für Programmierung und USART-Kommunikation integriert.

Die Entwicklerplatine AVR-USB-162 von Olimex übernimmt die Spannungsversorgung des Controllers, zusätzlich sind ein Reset-Button, ein Taster, eine LED und ein externer Quarz mit 8MHz im Lieferzustand enthalten. Die I/O-Pins sind auf Lötunkten herausgeführt und die Platine hat einen Lochrasterbereich, so kann sie leicht mit Bauteilen erweitert werden. Für dieses Projekt wurden zwei 7-Segmentanzeigen von Kingbright per Fädeldraht auf Port B, Port C und Port D angelötet, die genaue Belegung der Ports finden Sie unter Punkt 5.

### Software:

AVR-Studio 4  
Flip 3.4.1

## 3. Reaktionsspiel „Fang die Zahl“

Als Spiel wählte ich „Fang die Zahl“. Ziel des Reaktionsspiels ist es auf beiden 7-Segment-Anzeigen jeweils die gleichen Zahlen anzeigen zu lassen. Das Spiel endet bei 10 erfolgreichen Durchgängen, wobei sich die Schwierigkeit mit jedem Durchgang erhöht.

## 4. Spielablauf

Zu Beginn des Spiels laufen auf der Rechten 7-Segment-Anzeige die Zahlen zwischen 0-9 durch. Durch Betätigung des Tasters kann der Spieler eine Zahl wählen. Danach laufen auf der linken 7-Segment-Anzeige ebenfalls die Zahlen 0-9. Nun hat der Spieler die Aufgabe, durch Betätigung des Tasters, auf der linken 7-S-A die selbe Zahl zu „fangen“, wie diese auf der rechten 7-S-A angezeigt wird.

Hat der Spieler die richtige Zahl getroffen, wird die Anzahl der richtig durchlaufenen Durchgänge auf beiden 7-S-A angezeigt. Sollte der Spieler die falsche Zahl getroffen haben, wird der Buchstabe „F“ für Fehler auf beiden 7-S-A angezeigt. Nach jedem erfolgreichen Durchlauf, wird die Geschwindigkeit der durchlaufenden Zahlen erhöht, um die Schwierigkeit zu steigern.

Nach 10 erfolgreichen Durchgängen ist das Spiel beendet und die rote LED und die 7-S-A beginnen zu blinken. Das Spiel wird im Anschluss automatisch neu gestartet.



```

void zahl(int);
void entprellen(int);
int check_button(void);

int main(void)
{
    DDRB=0xff;
    DDRC=0xC0;
    DDRD= 0x7f;
    int i,gewl=0,gewr=0,anz_gew=0;           /* i: Zähler für for Schleifen,
                                           gewl, gewr: gewählte Zahl links bzw. rechts
                                           anz_gew: Anzahl der erfolgreichen Durchgänge */

    int z=1;                               /*Variable zur Überprüfung ob der Taster gedrückt wurde*/

    int anrz[12]= {0x77,0x14,0xB3,0xB6,0xD4,0xE6,0xE7,0x34,0xF7,0xF6,0x00,0xE1};
                                           /* Array für die Zeichen im rechten Display
                                           (0,1,2,3,4,5,6,7,8,9,löschen,F) */

    while(1)
    {
        while(anz_gew<10)
        {
            zahl(-1);
            z=1;
            while(z>0)                     /*Lasse Zahlen solange laufen, bis der Tastergedrückt wurde */
            {
                for(i=0;i<10;i++)          /*Zahlen auf dem rechten Display von 0 - 9 laufen lassen*/
                {
                    PORTB = anrz[i];
                    _delay_ms(DELAY);
                    if(check_button())
                    {
                        gewr=i;           /*Auswahl in gewr speichern*/
                        z=0;              /* Abbruchbedingung für while Schleife */
                        break;
                    }
                }
            }
            z=1;
            while(z>0)                     /*Lasse Zahlen solange laufen, bis der Tastergedrückt wurde */
            {
                for(i=0;i<10;i++)          /*Zahlen auf dem linken Display von 0 - 9 laufen
                                           lassen*/
                {
                    zahl(i);
                    _delay_ms(DELAY - (anz_gew*20));
                    if(check_button())
                    {
                        gewl=i;           /*Auswahl in gewl speichern*/
                        z=0;              /* Abbruchbedingung für while Schleife */
                        break;
                    }
                }
            }
            zahl(-1);                       /*7-S-Anzeige links löschen*/
            if(gewr==gewl)                  /*wenn linke Zahl gleich rechte Zahl*/
            {
                anz_gew++;                 /*erfolgreiche Durchgänge erhöhen*/
            }
        }
    }
}

```

```

        PORTB = anzr[anz_gew];      /*Anzeige der erfolgreichen Durchgänge auf
        zahl(anz_gew);              beiden 7-S-A anzeigen */
    }
    else                             /*linke Zahl ungleich rechte Zahl*/
    {
        zahl(11);                  /*Anzeige eines "F" auf beiden 7-S-A */
        PORTB = anzr[11];
    }
    _delay_ms(1500);                /*Pause für die Anzeige des F bzw. der erfolgreichen
    }                                  Durchgänge */
    anz_gew*=2;                      /* blinkt für jeden erfolgreichen Durchgang zwei mal*/

    PORTD ^= (1<<PORTD4);           /* LED anschalten */
    while(anz_gew>0)                /* Schleifen zum Blinken der Anzeigen und LED */
    {
        PORTD ^= (1<<PORTD4);       /* LED an/ausschalten */

        if(z==0)                   /* wenn 7-S-Anzeige aus, dann 0 anzeigen */
        {
            PORTB = anzr[0];
            zahl(0);
            z=1;
        }
        else                         /* wenn 7-S-Anzeige an, dann ausschalten */
        {
            PORTB =anzr[10];
            zahl(-1);
            z=0;
        }

        anz_gew--;
        _delay_ms(100);
    }
    PORTD &= ~(1<<PORTD4);          /* LED ausschalten */
    zahl(-1);                       /* linke 7-S-A löschen */
}
}

```

*/\* Diese Funktion setzt die benötigten LED's um die übergebene Zahl anzuzeigen  
Bei der Übergabe der Zahl 11: Der Buchstabe F wird angezeigt  
default: Anzeige wird nur gelöscht \*/*

```

void zahl(int i)
{
    PORTC &= ~((1 << PORTC6) | (1 << PORTC7));
    PORTD &= ~((1 << PORTD0) | (1 << PORTD1) | (1 << PORTD2) | (1 << PORTD3) | (1 << PORTD5) | (1
<< PORTD6));
    switch(i)
    {
        case 1:
            PORTD |= ((1 << PORTD6) | (1 << PORTD3));
            break;
        case 2:
            PORTD |= ((1 << PORTD2) | (1 << PORTD3) | (1 << PORTD0));
            PORTC |= ((1 << PORTC6) | (1 << PORTC7));
            break;
        case 3:

```

```

        PORTD |= ((1 << PORTD2) | (1 << PORTD3) | (1 << PORTD0) | (1 << PORTD6));
        PORTC |= ((1 << PORTC7));
        break;
    case 4:
        PORTD |= ((1 << PORTD1) | (1 << PORTD0) | (1 << PORTD3) | (1 << PORTD6));
        break;
    case 5:
        PORTD |= ((1 << PORTD2) | (1 << PORTD1) | (1 << PORTD0) | (1 << PORTD6));
        PORTC |= ((1 << PORTC7));
        break;
    case 6:
        PORTD |= ((1 << PORTD2) | (1 << PORTD1) | (1 << PORTD0) | (1 << PORTD6));
        PORTC |= ((1 << PORTC6) | (1 << PORTC7));
        break;
    case 7:
        PORTD |= ((1 << PORTD6) | (1 << PORTD3) | (1 << PORTD2));
        break;
    case 8:
        PORTC |= ((1 << PORTC6) | (1 << PORTC7));
        PORTD |= ((1 << PORTD0) | (1 << PORTD1) | (1 << PORTD2) | (1 << PORTD3) |
(1 << PORTD6));
        break;
    case 9:
        PORTD |= ((1 << PORTD2) | (1 << PORTD1) | (1 << PORTD0) | (1 << PORTD6) |
(1 << PORTD3));
        PORTC |= ((1 << PORTC7));
        break;
    case 0:
        PORTC |= ((1 << PORTC6) | (1 << PORTC7));
        PORTD |= ((1 << PORTD1) | (1 << PORTD2) | (1 << PORTD3) | (1 << PORTD6));
        break;
    case 11:
        PORTC |= ((1 << PORTC6) );
        PORTD |= ((1 << PORTD0) | (1 << PORTD1) | (1 << PORTD2));
        break;
    default:
        break;
}
}

```

*// In dieser Funktion wird überprüft ob der Taster gedrückt wurde. Ist dies der Fall, wird eine 1 zurückgegeben.*

```

int check_button()
{
    if(PIND&(1<<7))
    {
        return 0;
    }
    entprellen(7);
    while((PIND&(1<<7))==0);
    entprellen(7);
    return 1;
}

```

*// Funktion zum Entprellen des Schalters, falls dieser gedrückt wurde.*

```

void entprellen (int pin)
{
    int i=0;
    int current_state,last_state =0;
    while(i<ANZAHL_STABIL)
    {
        i++;
        current_state=(PIND&(1<<pin));
        if(current_state!=last_state)
        {
            i=0;
            last_state = current_state;
        }
    }
}

```

*/\*Copyright (c) 2010 Andreas Loy*

*Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:*

*The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.*

*THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.*

*Copyright © 2010 Andreas Loy*

*Hiermit wird unentgeltlich, jeder Person, die eine Kopie der Software und der zugehörigen Dokumentationen erhält, die Erlaubnis erteilt, diese uneingeschränkt zu benutzen, inklusive und ohne Ausnahme, dem Recht, sie zu verwenden, kopieren, ändern, fusionieren, verlegen, verbreiten, unterlizenzieren und/oder zu verkaufen, und Personen, die diese Software erhalten, diese Rechte zu geben, unter den folgenden Bedingungen:*

*Der obige Urheberrechtsvermerk und dieser Erlaubnisvermerk sind in alle Kopien oder Teilkopien der Software beizulegen.*

*DIE SOFTWARE WIRD OHNE JEDE AUSDRÜCKLICHE ODER IMPLIZIERTE GARANTIE BEREITGESTELLT, EINSCHLISSLICH DER GARANTIE ZUR BENUTZUNG FÜR DEN VORGESEHENEN ODER EINEM BESTIMMTEN ZWECK SOWIE JEDLICHER RECHTSVERLETZUNG, JEDOCH NICHT DARAUF BESCHRÄNKT. IN KEINEM FALL SIND DIE AUTOREN ODER COPYRIGHTINHABER FÜR JEDLICHEN SCHADEN ODER SONSTIGE ANSPRÜCHE HAFTBAR ZU MACHEN, OB INFOLGE DER ERFÜLLUNG EINES VERTRAGES, EINES DELIKTES ODER ANDERS IM ZUSAMMENHANG MIT DER SOFTWARE ODER SONSTIGER VERWENDUNG DER SOFTWARE ENTSTANDEN.\*//*